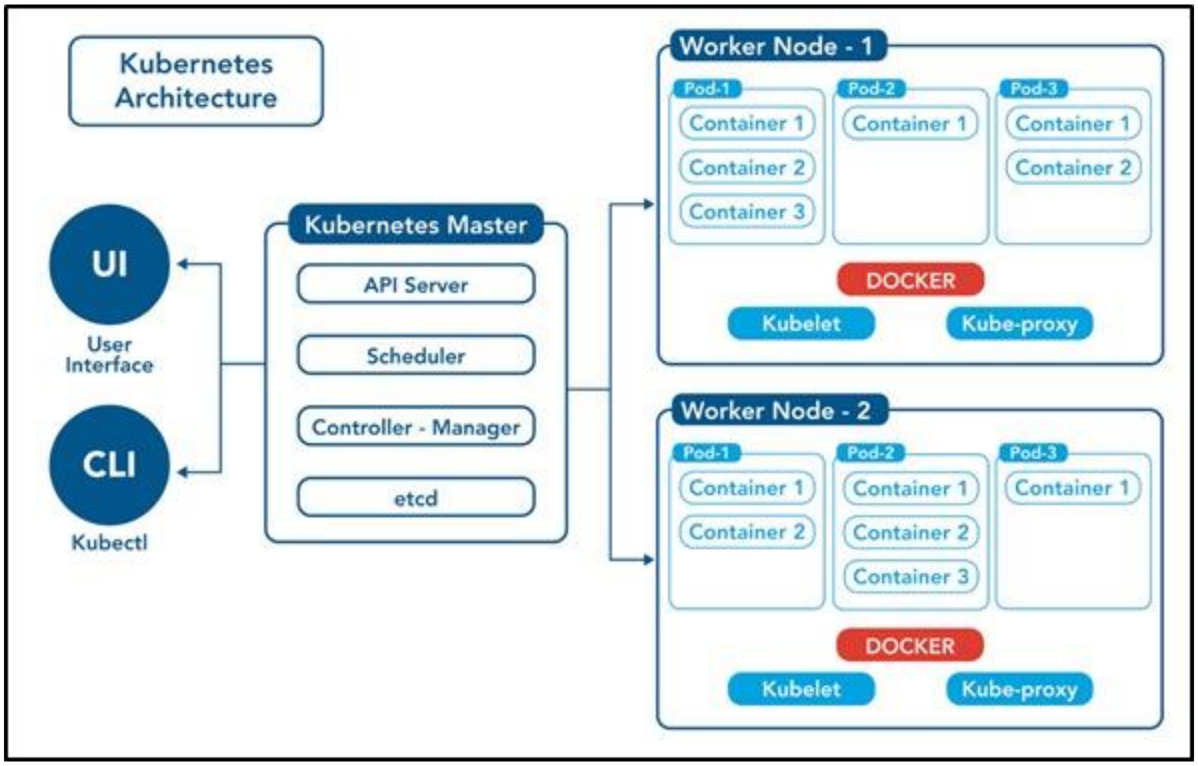
**Kubernetes (K8s)**

Kubernetes is an open-source container orchestration engine for automating deployment, scaling, and management of containerized applications. The open-source project is hosted by the Cloud Native Computing Foundation (CNCF).
It provides a scalable and resilient framework for automating the deployment, scaling, and management of applications across clusters of servers.



***A SMALL HISTORY OF K8S***:

-  In the early 2000s, Google started developing a system called Borg to manage their internal containerized applications.
-  Borg enabled Google to run applications at scale, providing features such as automatic scaling, service discovery, and fault tolerance.
-  In 2014, Google open-sourced a version of Borg called Kubernetes.
-  Kubernetes was donated to the Cloud Native Computing Foundation (CNCF), a neutral home for open-source cloud-native projects, in July 2015.
-  Kubernetes 1.8 added significant enhancements for storage, security, and networking. Key features included the stable release of the stateful sets API, expanded support for volume plugins, and improvements in security policies.

-  Check URL: https://kubernetes.io/releases/ for more release details.

**Control Plane /Master Node**
The control plane's components make global decisions about the cluster (for example, scheduling), as well as detecting and responding to cluster events (for example, starting up a new pod when a deployment's replicas field is unsatisfied).
Control plane components can be run on any machine in the cluster. Do not run user containers on this machine.
Node Components / Worker Nodes
Node components run on every node, maintaining running pods and providing the Kubernetes runtime environment.
1. Master Node: The master node is responsible for managing the cluster and coordinating the overall state of the system. It includes the following components:

a. API Server: The API server is the central control point for all interactions with the cluster. It exposes the Kubernetes API and handles requests from users and other components.

b. Scheduler: The scheduler is responsible for assigning workloads (pods) to individual worker nodes based on resource requirements, constraints, and other policies.

c. Controller Manager: The controller manager runs various controllers that monitor the cluster state and drive it towards the desired state. Examples include the replication controller, node controller, and service controller.

d. etcd: etcd is a distributed key-value store used by Kubernetes to store cluster state and configuration data.

1. **Pod**: The basic building block of Kubernetes. A pod represents a single instance of a running process within the cluster. It can encapsulate one or more containers that share the same network and storage resources.

1. **Create a pod using run command**

```
$ kubectl run <pod-name> --image=<image-name> --port=<container-port>
$ kubectl run my-pod --image=nginx --port=80
```

2. **View all the pods**
(In default namespace)

```
$ kubectl get pods
```

(In All namespace)

```
$ kubectl get pods -A
```

```
# For a specific namespace
$ kubectl get pods -n kube-system

# For a specific type
$ kubectl get pods <pod-name>
$ kubectl get pods <pod-name> -o wide
$ kubectl get pods <pod-name>  -o yaml
$ kubectl get pods <pod-name>  -o json
```

3. **Describe a pod (View Pod details**)

```
$ kubectl describe pod <pod-name>
$ kubectl describe pod my-pod
```

4. View Logs of a pod

```
$ kubectl logs <pod-name>
$ kubectl logs my-pod
```

**5. Execute any command inside Pod (Inside Pod OS)**

```
$ kubectl exec <pod-name> -- <command>
```

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  labels:
      app: my-web-app
    type: backend
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
        - containerPort: 80
```

```
apiVersion: v1
kind: Pod
metadata:
  name: my-app
spec:
  containers:
    - name: my-app-container
      image: <images>      ports:
        - containerPort: 9090
```

```
kind: ReplicaSet
metadata:
  name: my-rs
  labels:
    name: my-rs
spec:
  replicas: 4
  selector:
    matchLabels:
      apptype: web-backend
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image:
        ports:
          - containerPort: 8080
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 4
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image:
```

```
        ports:
              - containerPort: 7070
```

```
kubectl  create deployment webnginx2 --image=nginx:latest --replicas=1
```

```
kubectl scale deploy <deployment-name> --replicas=<desired-replica-count>
```

Services (short name = svc):
Service is an abstraction that defines a logical set of pods and a policy to access them. Services enable network connectivity and load balancing to the pods that are part of the service, allowing other components within or outside the cluster to interact with the application.
**Service Types: Kubernetes supports different types of services:**

1. **NodePort:** Exposes the service on a static port on each selected node's IP. This type makes the service accessible from outside the cluster by the <NodeIP>:<NodePort> combination.

2. **Cluster IP**: Exposes the service on a cluster-internal IP. This type makes the service only reachable within the cluster.

3. **LoadBalancer**: Creates an external load balancer in cloud environments, which routes traffic to the service.

**2. Create Deployment by executing above YAML file**

```
$ kubectl create -f web-deploy.yml
# Do necessary modifications if exist, else create new
$ kubectl create -f web-deploy.yml
# Completely Modify Pod Template
$ kubectl replace –f web-deploy.yml
```

**3. View Deployments**

```
$ kubectl get deployments
$ kubectl get deploy
$ kubectl get deploy -o wide
$ kubectl get deploy <deployment-name> -o json
$ kubectl get deploy <deployment-name> -o yaml
```

**4. View Deployment Description**

```
$ kubectl describe deploy <deployment-name>
```

**5. We can modify generated/updated YAML file**

```
$ kubectl edit deploy <deployment-name>
## change replicas: count to any other value then (ESC):wq
```

# We can modify our YAML file and then execute apply command
$ kubectl apply -f web-deploy.yml

## We can Even scale using command also
$ kubectl  scale  deploy  <deployment-name>   --replicas=<desired-replica-count>

**6. Delete Deploymen**t
$ kubectl delete deploy <deployment-name>
$ kubectl delete -f web-deploy.yml

**2. Create ReplicaSet by executing above YAML file**
$ kubectl create -f rs-test.yml
# Do necessary modifications if exist, else create new
$ kubectl apply -f rs-test.yml
# Completely Modify Pod Template
$ kubectl replace –f rs-test.yml

**3. View ReplicaSets**
$ kubectl get replicasets
$ kubectl get rs
$ kubectl get rs –o wide
$ kubectl get rs <replica-set-name> –o json
$ kubectl get rs <replica-set-name> –o yaml

4. **View ReplicaSet Descriptio**n
$ kubectl describe rs <replica-set-name>
5. We can modify generated/updated YAML file
$ kubectl edit rs <replica-set-name>
## change replicas: count to any other value then (ESC):wq

# We can modify our YAML file and then execute apply command
$ kubectl apply -f rs-test.yml

## We can Even scale using command also
$ kubectl scale replicaset <replicaset-name> --replicas=<desired-replica-count>

6. Delete ReplicaSet
$ kubectl delete rs <replica-set-name>
$ kubectl delete -f rs-test.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image:
        ports:
        - containerPort: 7070
---

apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
    type: backend-app
spec:
  type: NodePort
  ports:
   - targetPort: 7070
     port: 7070
     nodePort: 30002
  selector:
    apptype: web-backend
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image:
        ports:
        - containerPort: 9000

---

apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-service
spec:
  type: NodePort
  ports:
    - port: 9000
      targetPort: 8080
      nodePort: 30002
  selector:
    apptype: web-backend
```

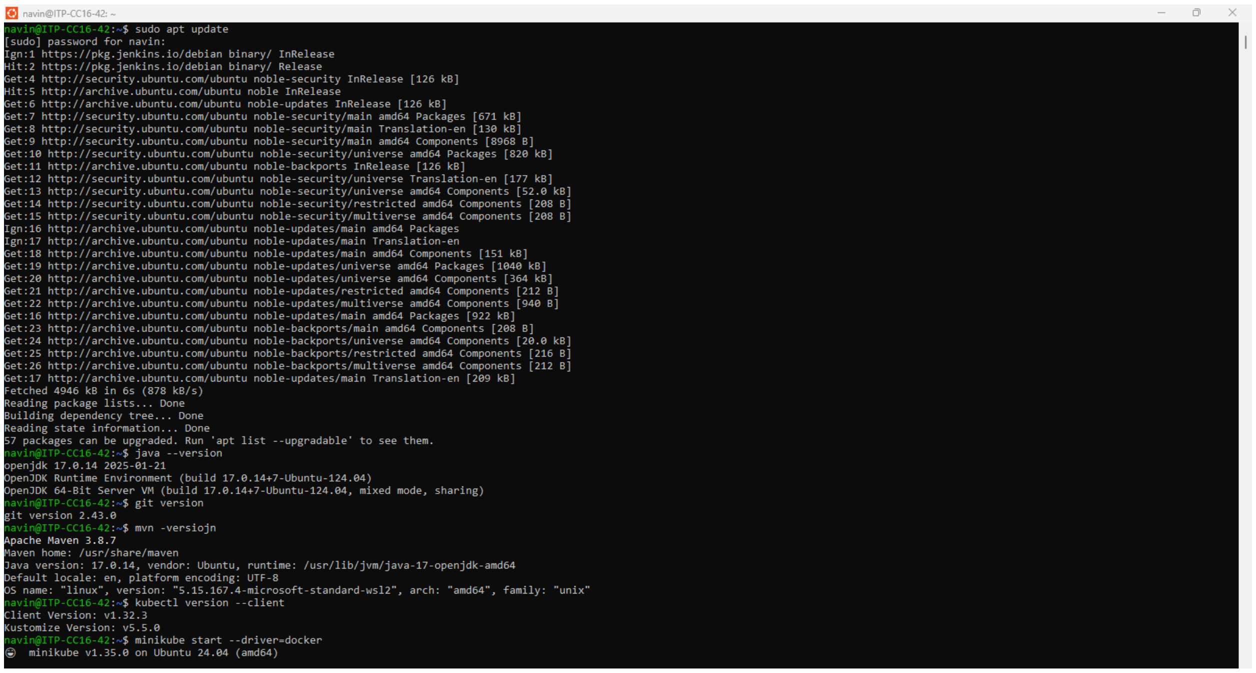
Namespace (short name = ns):
namespace is a virtual cluster or logical partition within a cluster that provides a way to organize and isolate resources. It allows multiple teams or projects to share the same physical cluster while maintaining resource separation and access control.

```
# To create a namespace:
$ kubectl create namespace <namespace-name>
$ kubectl create ns my-bank
# To switch to a specific namespace: (make this as default type)
$ kubectl config set-context --current --namespace=<namespace-name>
# To list all namespaces:
$ kubectl get namespaces
# To get resources within a specific namespace:
$ kubectl get <resource-type> -n <namespace-name>
$ kubectl get deploy -n my-bank
$ kubectl get deploy --namespace my-bank
$ kubectl get all --namespace my-bank
# To delete a namespace and all associated resources:
$ kubectl delete namespace <namespace-name>
$ kubectl delete ns my-bank
```

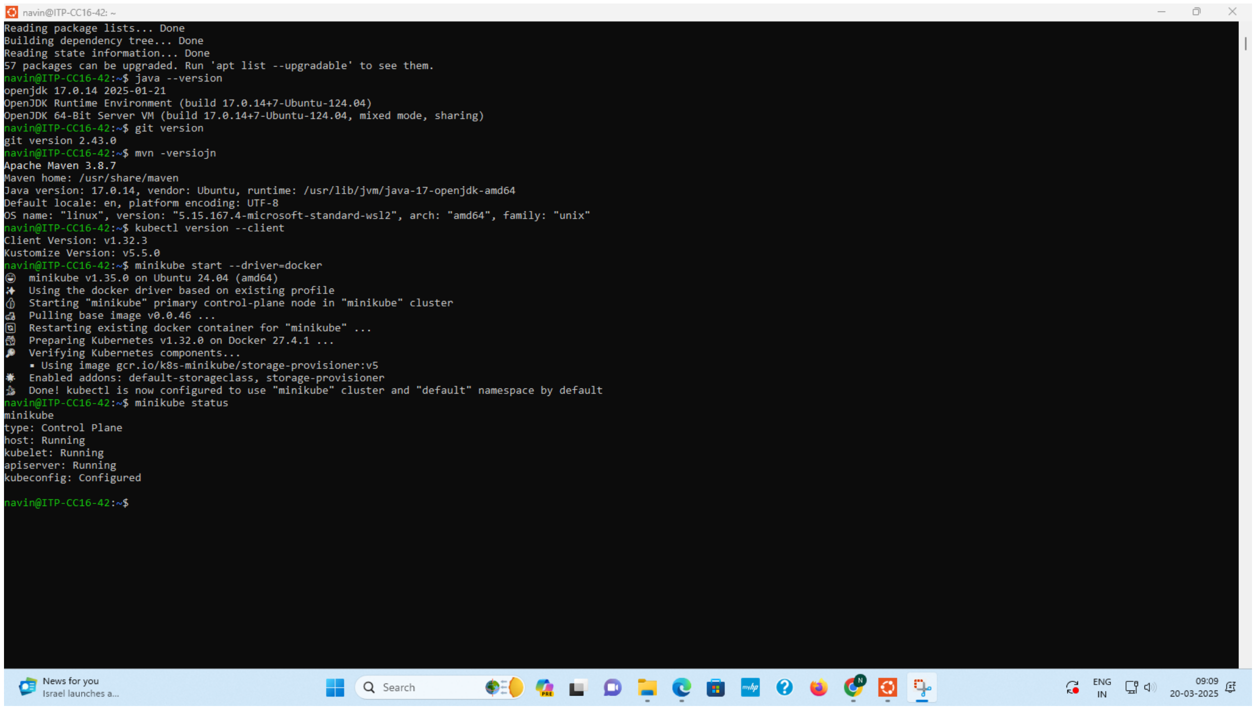
```
kubectl create ns mydeploy
kubectl apply -f deploy.yml -n mydeploy
```

```
apiVersion: v1
kind: Namespace
metadata:
  name: my-demo-ns
```

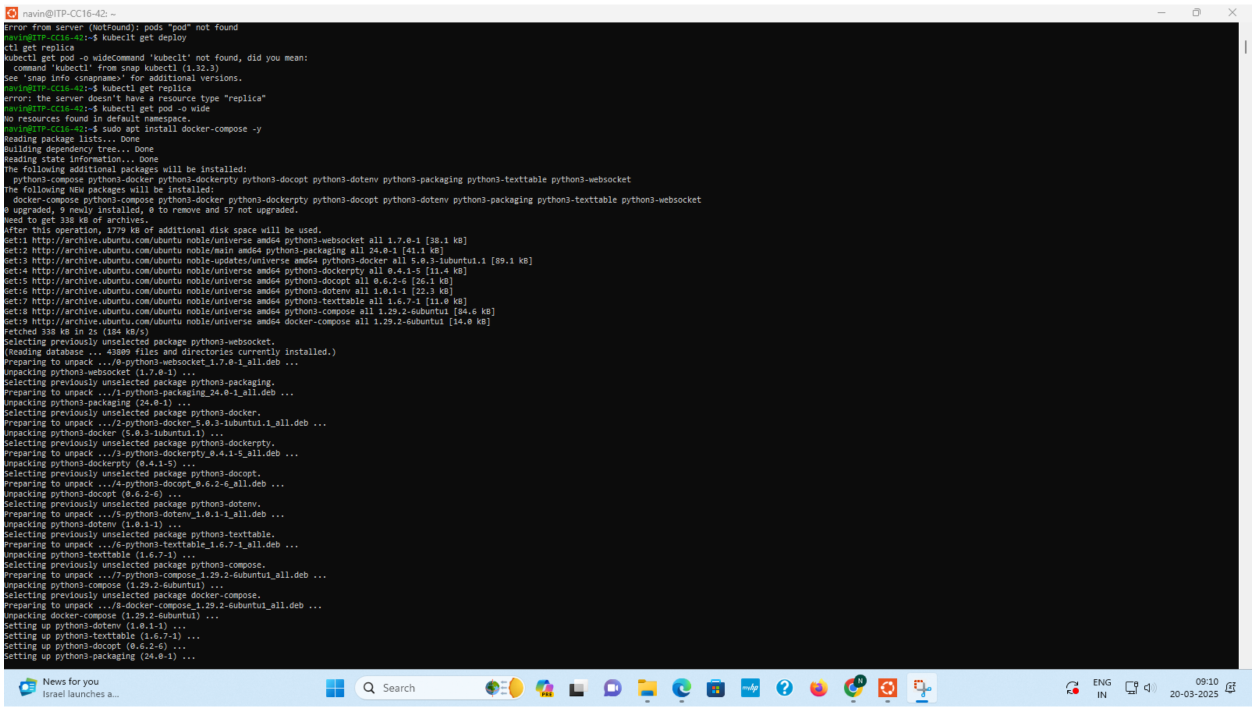
```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  namespace: my-demo-ns
spec:
  containers:
  - name: my-container
    image: nginx:latest
```



```
navin@ITP-CC16-42:~$ sudo apt update
[sudo] password for navin:
Ign:1 https://pkg.jenkins.io/debian binary/ InRelease
Hit:2 https://pkg.jenkins.io/debian binary/ Release
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:5 http://archive.ubuntu.com/ubuntu noble InRelease
Get:6 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [671 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [130 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8968 B]
Get:10 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [820 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [177 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.0 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:15 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Ign:16 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages
Ign:17 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en
Get:18 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1040 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:16 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [922 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 B]
Get:24 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [20.0 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:26 http://archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:17 http://archive.ubuntu.com/ubuntu noble-updates/main Translation-en [209 kB]
Fetched 4946 kB in 6s (878 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
57 packages can be upgraded. Run 'apt list --upgradable' to see them.
navin@ITP-CC16-42:~$ java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-124.04, mixed mode, sharing)
navin@ITP-CC16-42:~$ git version
git version 2.43.0
navin@ITP-CC16-42:~$ mvn -versiojn
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 17.0.14, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.15.167.4-microsoft-standard-wsl2", arch: "amd64", family: "unix"
navin@ITP-CC16-42:~$ kubectl version --client
Client Version: v1.32.3
Kustomize Version: v5.5.0
navin@ITP-CC16-42:~$ minikube start --driver=docker
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
```

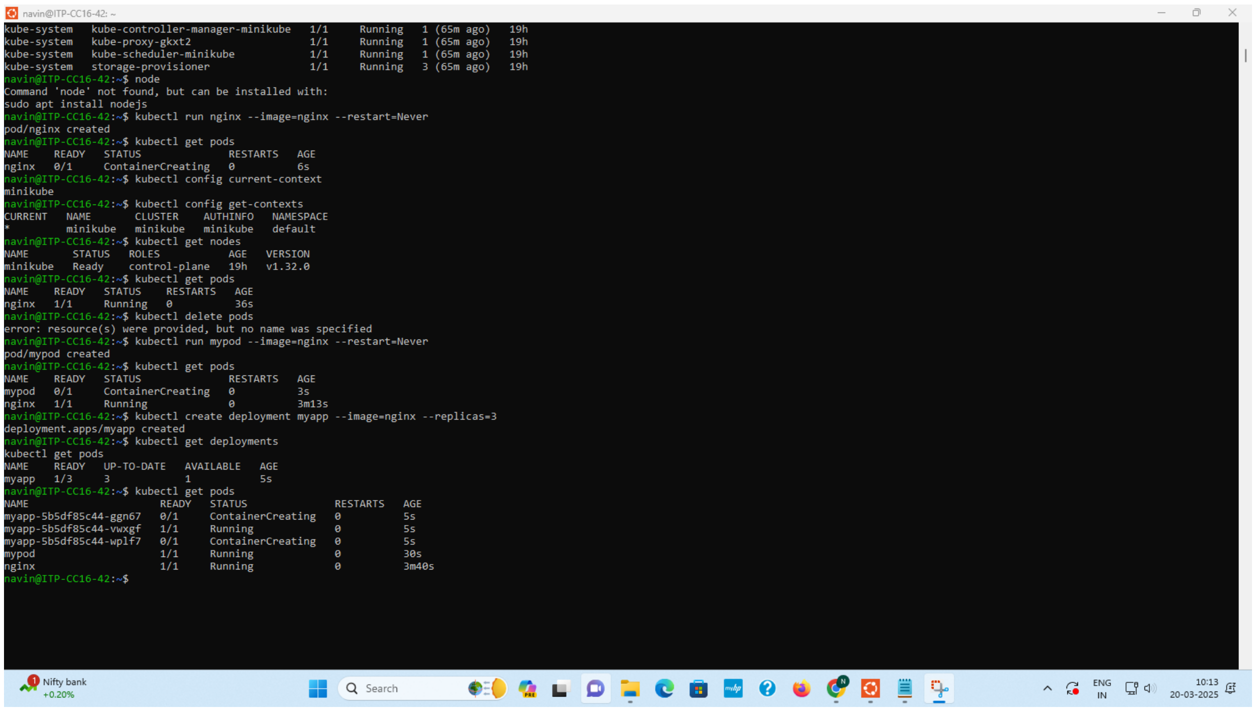


```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
57 packages can be upgraded. Run 'apt list --upgradable' to see them.
navin@ITP-CC16-42:~$ java --version
openjdk 17.0.14 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-124.04, mixed mode, sharing)
navin@ITP-CC16-42:~$ git version
git version 2.43.0
navin@ITP-CC16-42:~$ mvn -versiojn
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 17.0.14, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.15.167.4-microsoft-standard-wsl2", arch: "amd64", family: "unix"
navin@ITP-CC16-42:~$ kubectl version --client
Client Version: v1.32.3
Kustomize Version: v5.5.0
navin@ITP-CC16-42:~$ minikube start --driver=docker
😄  minikube v1.35.0 on Ubuntu 24.04 (amd64)
✨  Using the docker driver based on existing profile
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
navin@ITP-CC16-42:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

navin@ITP-CC16-42:~$
```



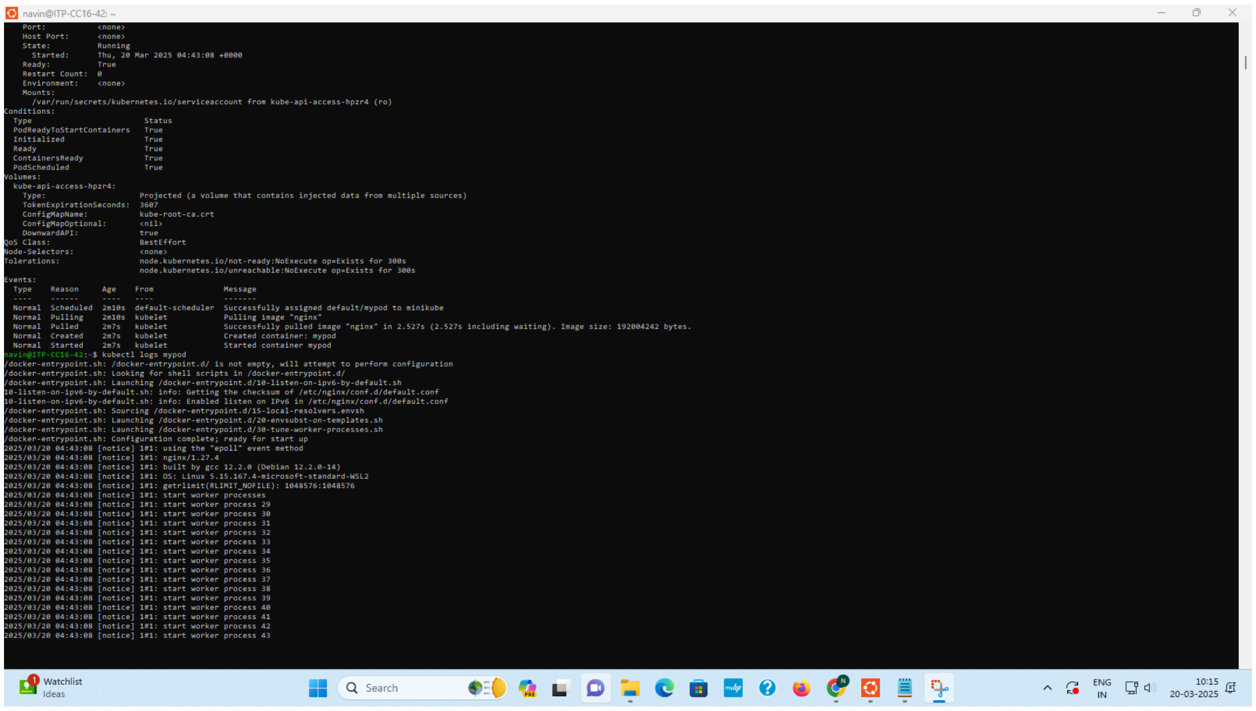
navin@ITP-CC16-42: ~

```
Error from server (NotFound): pods "pod" not found
navin@ITP-CC16-42:~$ kubeclt get deploy
ctl get replica
kubectl get pod -o wideCommand 'kubeclt' not found, did you mean:
  command 'kubectl' from snap kubectl (1.32.3)
See 'snap info <snapname>' for additional versions.
navin@ITP-CC16-42:~$ kubectl get replica
error: the server doesn't have a resource type "replica"
navin@ITP-CC16-42:~$ kubectl get pod -o wide
No resources found in default namespace.
navin@ITP-CC16-42:~$ sudo apt install docker-compose -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-compose python3-docker python3-dockerpty python3-docopt python3-dotenv python3-packaging python3-texttable python3-websocket
The following NEW packages will be installed:
  docker-compose python3-compose python3-docker python3-dockerpty python3-docopt python3-dotenv python3-packaging python3-texttable python3-websocket
0 upgraded, 9 newly installed, 0 to remove and 57 not upgraded.
Need to get 338 kB of archives.
After this operation, 1779 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-websocket all 1.7.0-1 [38.1 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/main amd64 python3-packaging all 24.0-1 [41.1 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 python3-docker all 5.0.3-1ubuntu1.1 [89.1 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-dockerpty all 0.4.1-5 [11.4 kB]
Get:5 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-docopt all 0.6.2-6 [26.1 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-dotenv all 1.0.1-1 [22.3 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-texttable all 1.6.7-1 [11.0 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-compose all 1.29.2-6ubuntu1 [84.6 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 docker-compose all 1.29.2-6ubuntu1 [14.0 kB]
Fetched 338 kB in 2s (184 kB/s)
Selecting previously unselected package python3-websocket.
(Reading database ... 43809 files and directories currently installed.)
Preparing to unpack .../0-python3-websocket_1.7.0-1_all.deb ...
Unpacking python3-websocket (1.7.0-1) ...
Selecting previously unselected package python3-packaging.
Preparing to unpack .../1-python3-packaging_24.0-1_all.deb ...
Unpacking python3-packaging (24.0-1) ...
Selecting previously unselected package python3-docker.
Preparing to unpack .../2-python3-docker_5.0.3-1ubuntu1.1_all.deb ...
Unpacking python3-docker (5.0.3-1ubuntu1.1) ...
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../3-python3-dockerpty_0.4.1-5_all.deb ...
Unpacking python3-dockerpty (0.4.1-5) ...
Selecting previously unselected package python3-docopt.
Preparing to unpack .../4-python3-docopt_0.6.2-6_all.deb ...
Unpacking python3-docopt (0.6.2-6) ...
Selecting previously unselected package python3-dotenv.
Preparing to unpack .../5-python3-dotenv_1.0.1-1_all.deb ...
Unpacking python3-dotenv (1.0.1-1) ...
Selecting previously unselected package python3-texttable.
Preparing to unpack .../6-python3-texttable_1.6.7-1_all.deb ...
Unpacking python3-texttable (1.6.7-1) ...
Selecting previously unselected package python3-compose.
Preparing to unpack .../7-python3-compose_1.29.2-6ubuntu1_all.deb ...
Unpacking python3-compose (1.29.2-6ubuntu1) ...
Selecting previously unselected package docker-compose.
Preparing to unpack .../8-docker-compose_1.29.2-6ubuntu1_all.deb ...
Unpacking docker-compose (1.29.2-6ubuntu1) ...
Setting up python3-dotenv (1.0.1-1) ...
Setting up python3-texttable (1.6.7-1) ...
Setting up python3-docopt (0.6.2-6) ...
Setting up python3-packaging (24.0-1) ...
```



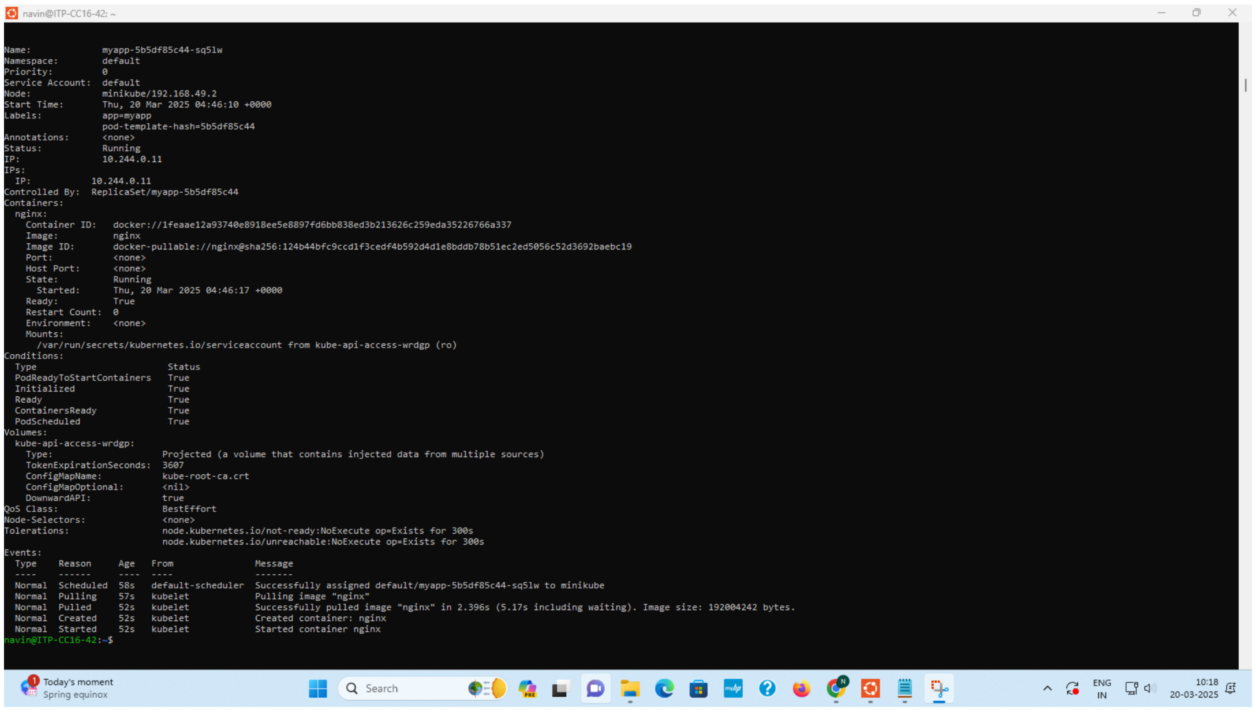
navin@ITP-CC16-42: ~

```
kube-system   kube-controller-manager-minikube   1/1     Running   1 (65m ago)   19h
kube-system   kube-proxy-gkxt2                   1/1     Running   1 (65m ago)   19h
kube-system   kube-scheduler-minikube            1/1     Running   1 (65m ago)   19h
kube-system   storage-provisioner                1/1     Running   3 (65m ago)   19h
navin@ITP-CC16-42:~$ node
Command 'node' not found, but can be installed with:
sudo apt install nodejs
navin@ITP-CC16-42:~$ kubectl run nginx --image=nginx --restart=Never
pod/nginx created
navin@ITP-CC16-42:~$ kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
nginx   0/1     ContainerCreating   0          6s
navin@ITP-CC16-42:~$ kubectl config current-context
minikube
navin@ITP-CC16-42:~$ kubectl config get-contexts
CURRENT   NAME       CLUSTER    AUTHINFO   NAMESPACE
*         minikube   minikube   minikube   default
navin@ITP-CC16-42:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   19h   v1.32.0
navin@ITP-CC16-42:~$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          36s
navin@ITP-CC16-42:~$ kubectl delete pods
error: resource(s) were provided, but no name was specified
navin@ITP-CC16-42:~$ kubectl run mypod --image=nginx --restart=Never
pod/mypod created
navin@ITP-CC16-42:~$ kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
mypod   0/1     ContainerCreating   0          3s
nginx   1/1     Running             0          3m13s
navin@ITP-CC16-42:~$ kubectl create deployment myapp --image=nginx --replicas=3
deployment.apps/myapp created
navin@ITP-CC16-42:~$ kubectl get deployments
kubectl get pods
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
myapp   1/3     3            1           5s
navin@ITP-CC16-42:~$ kubectl get pods
NAME                     READY   STATUS              RESTARTS   AGE
myapp-5b5df85c44-ggn67   0/1     ContainerCreating   0          5s
myapp-5b5df85c44-vwxgf   1/1     Running             0          5s
myapp-5b5df85c44-wplf7   0/1     ContainerCreating   0          5s
mypod                    1/1     Running             0          30s
nginx                    1/1     Running             0          3m40s
navin@ITP-CC16-42:~$
```



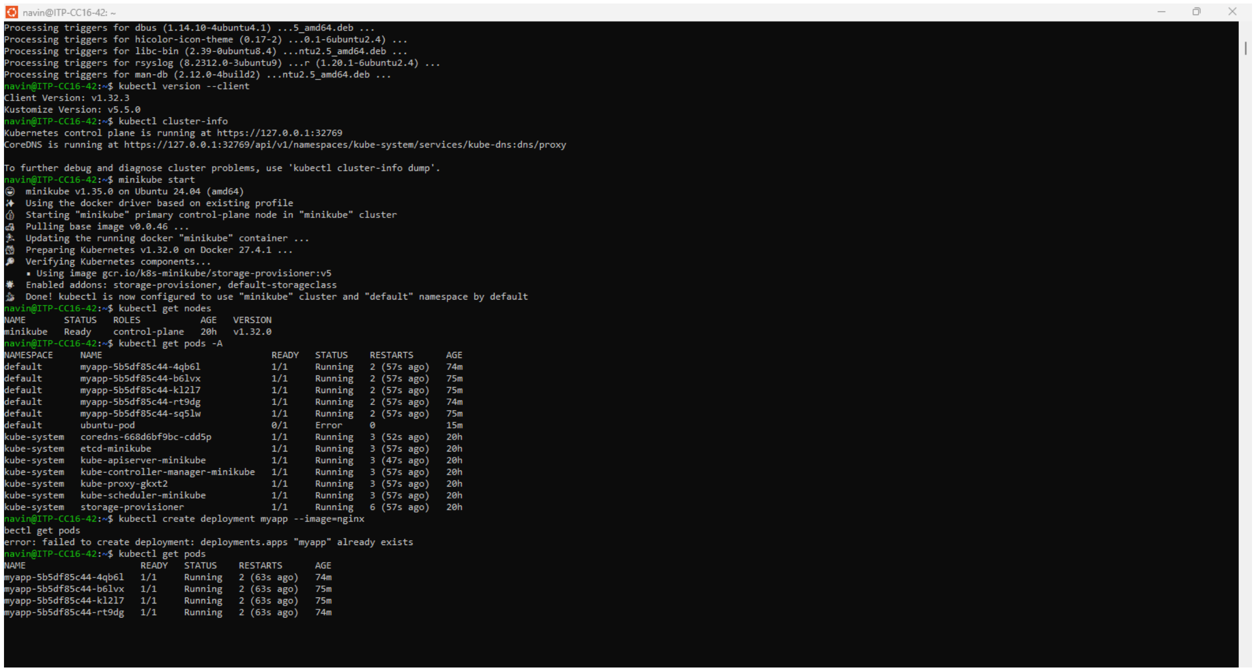
navin@ITP-CC16-42: ~

```
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 20 Mar 2025 04:43:08 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-hpzr4 (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-hpzr4:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age    From               Message
  ----    ------     ----   ----               -------
  Normal  Scheduled  2m10s  default-scheduler  Successfully assigned default/mypod to minikube
  Normal  Pulling    2m10s  kubelet            Pulling image "nginx"
  Normal  Pulled     2m7s   kubelet            Successfully pulled image "nginx" in 2.527s (2.527s including waiting). Image size: 192004242 bytes.
  Normal  Created    2m7s   kubelet            Created container: mypod
  Normal  Started    2m7s   kubelet            Started container mypod
navin@ITP-CC16-42:~$ kubectl logs mypod
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/03/20 04:43:08 [notice] 1#1: using the "epoll" event method
2025/03/20 04:43:08 [notice] 1#1: nginx/1.27.4
2025/03/20 04:43:08 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/03/20 04:43:08 [notice] 1#1: OS: Linux 5.15.167.4-microsoft-standard-WSL2
2025/03/20 04:43:08 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/03/20 04:43:08 [notice] 1#1: start worker processes
2025/03/20 04:43:08 [notice] 1#1: start worker process 29
2025/03/20 04:43:08 [notice] 1#1: start worker process 30
2025/03/20 04:43:08 [notice] 1#1: start worker process 31
2025/03/20 04:43:08 [notice] 1#1: start worker process 32
2025/03/20 04:43:08 [notice] 1#1: start worker process 33
2025/03/20 04:43:08 [notice] 1#1: start worker process 34
2025/03/20 04:43:08 [notice] 1#1: start worker process 35
2025/03/20 04:43:08 [notice] 1#1: start worker process 36
2025/03/20 04:43:08 [notice] 1#1: start worker process 37
2025/03/20 04:43:08 [notice] 1#1: start worker process 38
2025/03/20 04:43:08 [notice] 1#1: start worker process 39
2025/03/20 04:43:08 [notice] 1#1: start worker process 40
2025/03/20 04:43:08 [notice] 1#1: start worker process 41
2025/03/20 04:43:08 [notice] 1#1: start worker process 42
2025/03/20 04:43:08 [notice] 1#1: start worker process 43
```

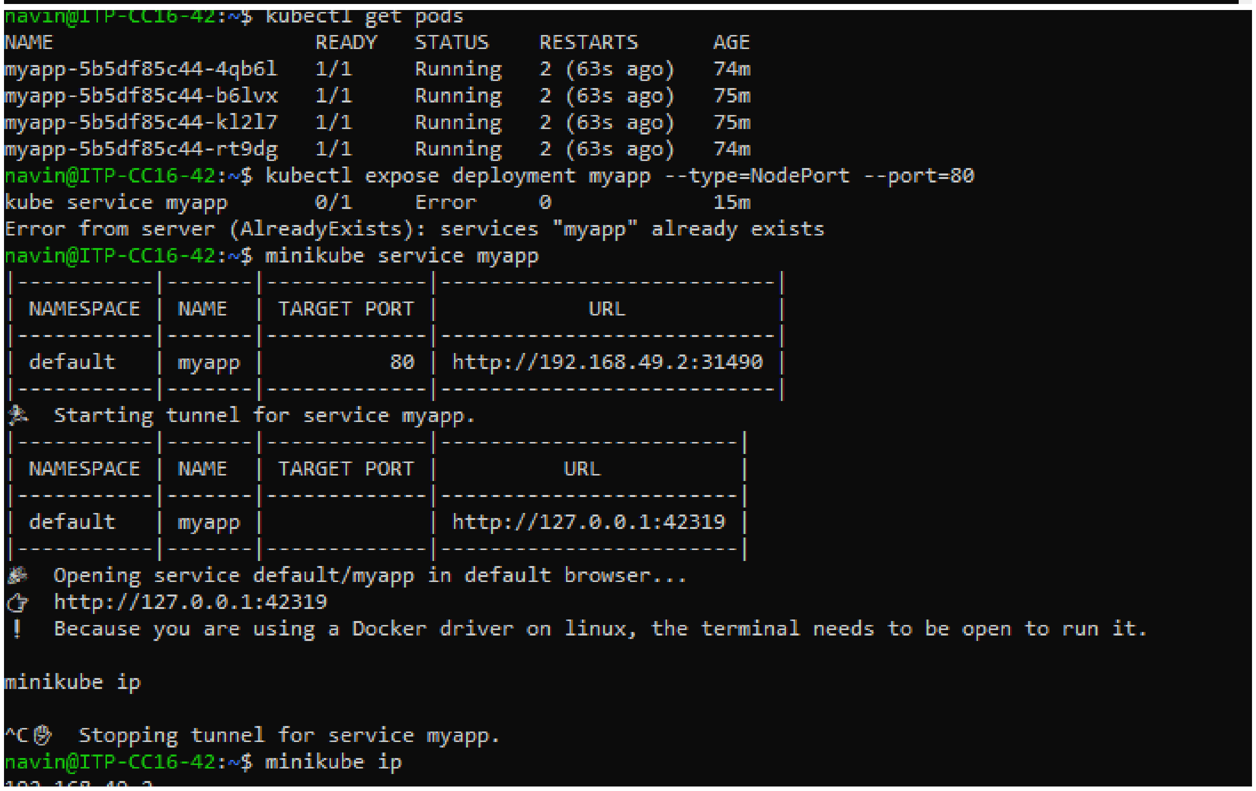


navin@ITP-CC16-42: ~

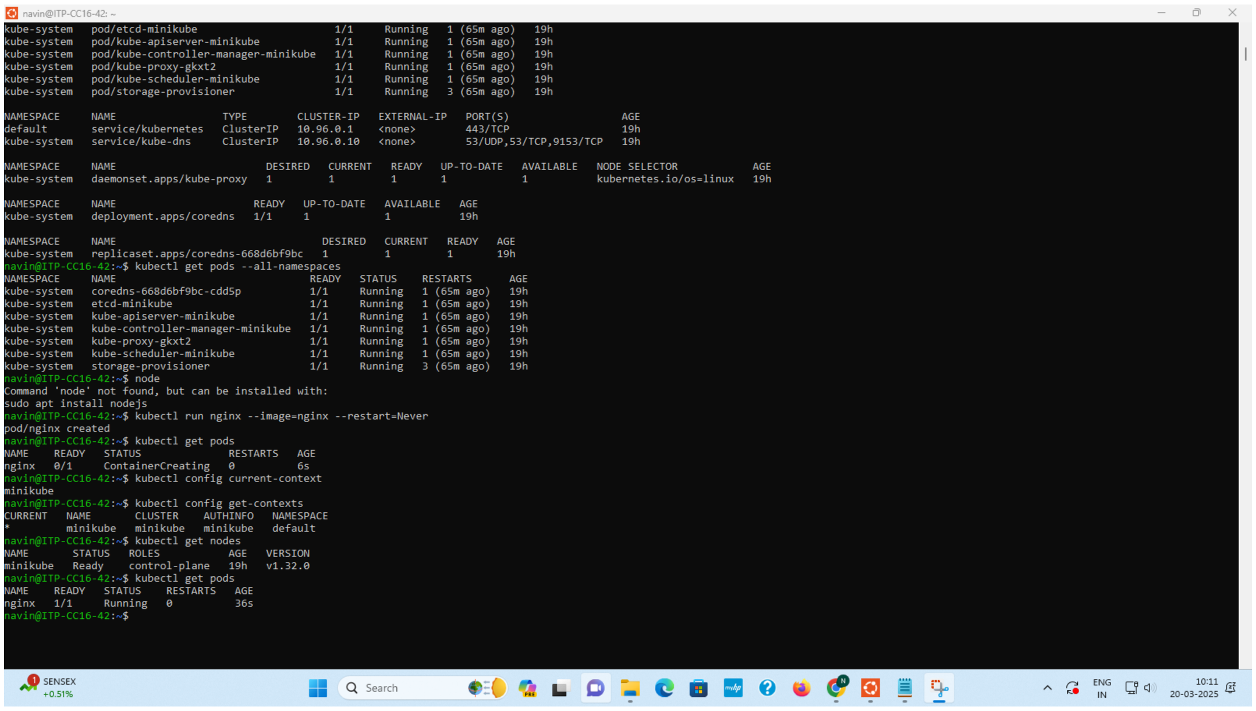
```
Name:             myapp-5b5df85c44-sq5lw
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Thu, 20 Mar 2025 04:46:10 +0000
Labels:           app=myapp
                  pod-template-hash=5b5df85c44
Annotations:      <none>
Status:           Running
IP:               10.244.0.11
IPs:
  IP:           10.244.0.11
Controlled By:  ReplicaSet/myapp-5b5df85c44
Containers:
  nginx:
    Container ID:   docker://1feaae12a93740e8918ee5e8897fd6bb838ed3b213626c259eda35226766a337
    Image:          nginx
    Image ID:       docker-pullable://nginx@sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 20 Mar 2025 04:46:17 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-wrdgp (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-wrdgp:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  58s   default-scheduler  Successfully assigned default/myapp-5b5df85c44-sq5lw to minikube
  Normal  Pulling    57s   kubelet            Pulling image "nginx"
  Normal  Pulled     52s   kubelet            Successfully pulled image "nginx" in 2.396s (5.17s including waiting). Image size: 192004242 bytes.
  Normal  Created    52s   kubelet            Created container: nginx
  Normal  Started    52s   kubelet            Started container nginx
navin@ITP-CC16-42:~$
```



```
navin@ITP-CC16-42: ~
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...5_amd64.deb ...
Processing triggers for hicolor-icon-theme (0.17-2) ...0.1-6ubuntu2.4) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...ntu2.5_amd64.deb ...
Processing triggers for rsyslog (8.2312.0-3ubuntu9) ...r (1.20.1-6ubuntu2.4) ...
Processing triggers for man-db (2.12.0-4build2) ...ntu2.5_amd64.deb ...
navin@ITP-CC16-42:~$ kubectl version --client
Client Version: v1.32.3
Kustomize Version: v5.5.0
navin@ITP-CC16-42:~$ kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:32769
CoreDNS is running at https://127.0.0.1:32769/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
navin@ITP-CC16-42:~$ minikube start
   minikube v1.35.0 on Ubuntu 24.04 (amd64)
   Using the docker driver based on existing profile
   Starting "minikube" primary control-plane node in "minikube" cluster
   Pulling base image v0.0.46 ...
   Updating the running docker "minikube" container ...
   Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
   Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
   Enabled addons: storage-provisioner, default-storageclass
   Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
navin@ITP-CC16-42:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   20h   v1.32.0
navin@ITP-CC16-42:~$ kubectl get pods -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS      AGE
default       myapp-5b5df85c44-4qb6l             1/1     Running   2 (57s ago)   74m
default       myapp-5b5df85c44-b6lvx             1/1     Running   2 (57s ago)   75m
default       myapp-5b5df85c44-kl2l7             1/1     Running   2 (57s ago)   75m
default       myapp-5b5df85c44-rt9dg             1/1     Running   2 (57s ago)   74m
default       myapp-5b5df85c44-sq5lw             1/1     Running   2 (57s ago)   75m
default       ubuntu-pod                         0/1     Error     0             15m
kube-system   coredns-668d6bf9bc-cdd5p           1/1     Running   3 (52s ago)   20h
kube-system   etcd-minikube                      1/1     Running   3 (57s ago)   20h
kube-system   kube-apiserver-minikube            1/1     Running   3 (47s ago)   20h
kube-system   kube-controller-manager-minikube   1/1     Running   3 (57s ago)   20h
kube-system   kube-proxy-gkxt2                   1/1     Running   3 (57s ago)   20h
kube-system   kube-scheduler-minikube            1/1     Running   3 (57s ago)   20h
kube-system   storage-provisioner                1/1     Running   6 (57s ago)   20h
navin@ITP-CC16-42:~$ kubectl create deployment myapp --image=nginx
bectl get pods
error: failed to create deployment: deployments.apps "myapp" already exists
navin@ITP-CC16-42:~$ kubectl get pods
NAME                     READY   STATUS    RESTARTS      AGE
myapp-5b5df85c44-4qb6l   1/1     Running   2 (63s ago)   74m
myapp-5b5df85c44-b6lvx   1/1     Running   2 (63s ago)   75m
myapp-5b5df85c44-kl2l7   1/1     Running   2 (63s ago)   75m
myapp-5b5df85c44-rt9dg   1/1     Running   2 (63s ago)   74m
```



```
navin@ITP-CC16-42:~$ kubectl get pods
NAME                     READY   STATUS    RESTARTS      AGE
myapp-5b5df85c44-4qb6l   1/1     Running   2 (63s ago)   74m
myapp-5b5df85c44-b6lvx   1/1     Running   2 (63s ago)   75m
myapp-5b5df85c44-kl2l7   1/1     Running   2 (63s ago)   75m
myapp-5b5df85c44-rt9dg   1/1     Running   2 (63s ago)   74m
navin@ITP-CC16-42:~$ kubectl expose deployment myapp --type=NodePort --port=80
kube service myapp       0/1     Error     0             15m
Error from server (AlreadyExists): services "myapp" already exists
navin@ITP-CC16-42:~$ minikube service myapp
|-----------|-------|-------------|---------------------------|
| NAMESPACE | NAME  | TARGET PORT |            URL            |
|-----------|-------|-------------|---------------------------|
| default   | myapp |          80 | http://192.168.49.2:31490 |
|-----------|-------|-------------|---------------------------|
   Starting tunnel for service myapp.
|-----------|-------|-------------|------------------------|
| NAMESPACE | NAME  | TARGET PORT |          URL           |
|-----------|-------|-------------|------------------------|
| default   | myapp |             | http://127.0.0.1:42319 |
|-----------|-------|-------------|------------------------|
   Opening service default/myapp in default browser...
   http://127.0.0.1:42319
!  Because you are using a Docker driver on linux, the terminal needs to be open to run it.

minikube ip

^C   Stopping tunnel for service myapp.
navin@ITP-CC16-42:~$ minikube ip
```



```
kube-system   pod/etcd-minikube                      1/1     Running   1 (65m ago)   19h
kube-system   pod/kube-apiserver-minikube            1/1     Running   1 (65m ago)   19h
kube-system   pod/kube-controller-manager-minikube   1/1     Running   1 (65m ago)   19h
kube-system   pod/kube-proxy-gkxt2                   1/1     Running   1 (65m ago)   19h
kube-system   pod/kube-scheduler-minikube            1/1     Running   1 (65m ago)   19h
kube-system   pod/storage-provisioner                1/1     Running   3 (65m ago)   19h

NAMESPACE     NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)                  AGE
default       service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP                  19h
kube-system   service/kube-dns     ClusterIP   10.96.0.10   <none>        53/UDP,53/TCP,9153/TCP   19h

NAMESPACE     NAME                        DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
kube-system   daemonset.apps/kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   19h

NAMESPACE     NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
kube-system   deployment.apps/coredns   1/1     1            1           19h

NAMESPACE     NAME                                 DESIRED   CURRENT   READY   AGE
kube-system   replicaset.apps/coredns-668d6bf9bc   1         1         1       19h
navin@ITP-CC16-42:~$ kubectl get pods --all-namespaces
NAMESPACE     NAME                               READY   STATUS    RESTARTS      AGE
kube-system   coredns-668d6bf9bc-cdd5p           1/1     Running   1 (65m ago)   19h
kube-system   etcd-minikube                      1/1     Running   1 (65m ago)   19h
kube-system   kube-apiserver-minikube            1/1     Running   1 (65m ago)   19h
kube-system   kube-controller-manager-minikube   1/1     Running   1 (65m ago)   19h
kube-system   kube-proxy-gkxt2                   1/1     Running   1 (65m ago)   19h
kube-system   kube-scheduler-minikube            1/1     Running   1 (65m ago)   19h
kube-system   storage-provisioner                1/1     Running   3 (65m ago)   19h
navin@ITP-CC16-42:~$ node
Command 'node' not found, but can be installed with:
sudo apt install nodejs
navin@ITP-CC16-42:~$ kubectl run nginx --image=nginx --restart=Never
pod/nginx created
navin@ITP-CC16-42:~$ kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
nginx   0/1     ContainerCreating   0          6s
navin@ITP-CC16-42:~$ kubectl config current-context
minikube
navin@ITP-CC16-42:~$ kubectl config get-contexts
CURRENT   NAME       CLUSTER    AUTHINFO   NAMESPACE
*         minikube   minikube   minikube   default
navin@ITP-CC16-42:~$ kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   19h   v1.32.0
navin@ITP-CC16-42:~$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
nginx   1/1     Running   0          36s
navin@ITP-CC16-42:~$
```